

Phân Tích Dữ Liệu Thực Tế với Python

Bài 12.1: Lập trình Hướng đối tượng trong Python



Quang-Khai Tran, Ph.D
CyberLab, 04/2023



(Ảnh: Internet)

Nội dung



1. Giới thiệu OOP trong Python
2. Demo: blockchain
3. Thảo Luận

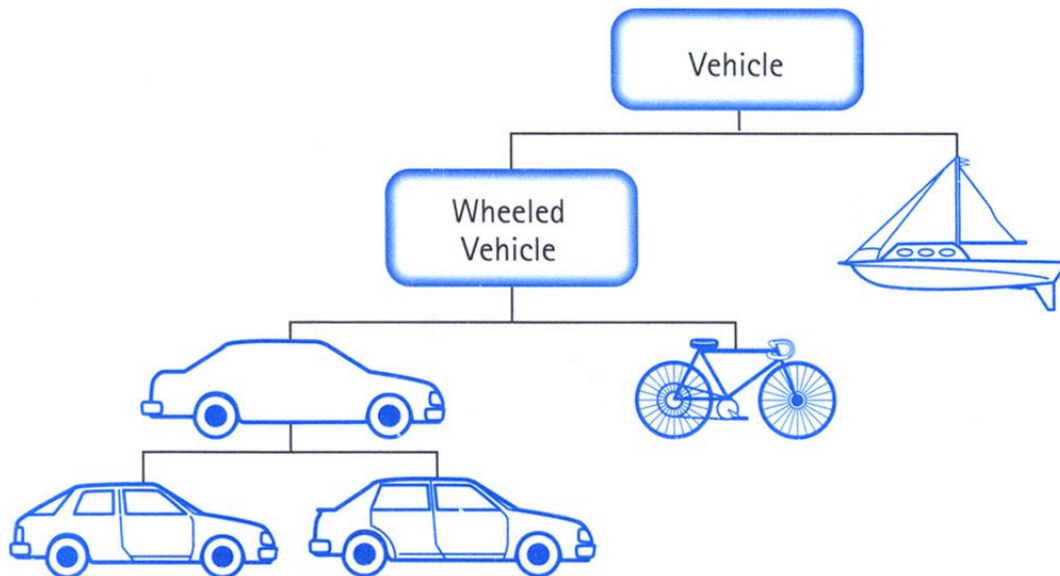
Lập Trình Hướng Đối Tượng (Object-Oriented Programming) là gì?

"Là phương pháp tổ chức cấu trúc một chương trình bằng cách bọc/bó các thuộc tính và hoạt động liên quan với nhau thành các đối tượng riêng biệt"



Source: <https://realpython.com/python3-object-oriented-programming/>

Ví dụ:

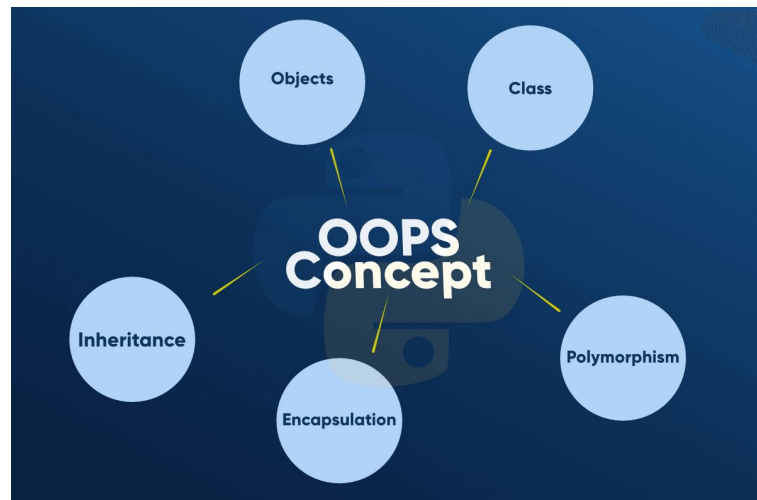


Source:

<https://www.java67.com/2016/09/oops-concept-tutorial-in-java-object-oriented-programming.html>

Một số khái niệm chính của OO:

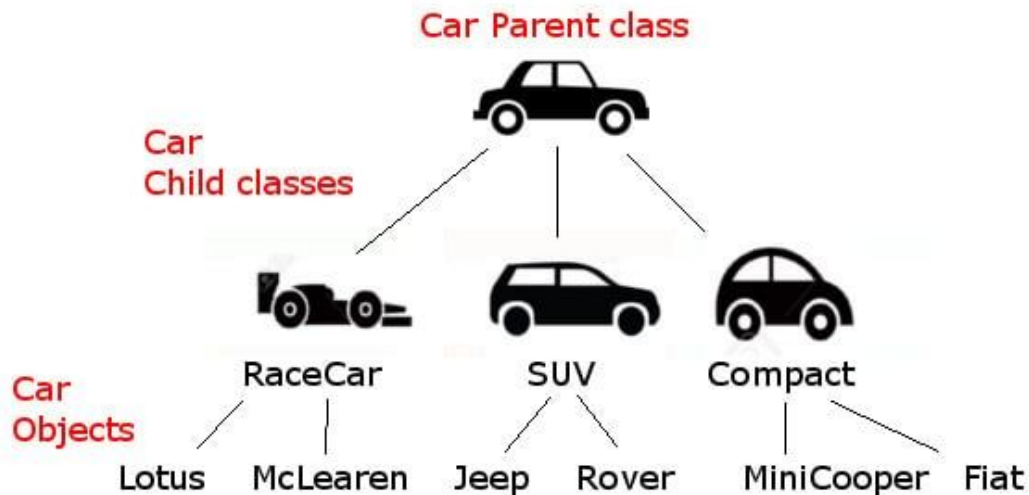
- ❖ Objects (đối tượng)
 - ❖ Class (lớp)
 - ❖ Instance (một thể hiện)
 - ❖ Inheritance (kế thừa)
 - ❖ Encapsulation (đóng gói)
 - ❖ Polymorphism (đa xạ)
-
- ❖ Tham khảo:
 - https://www.tutorialspoint.com/python/python_classes_objects.htm
 - <https://realpython.com/python3-object-oriented-programming/>
 - <https://viblo.asia/p/oop-voi-python-E375zQGblGW> (tiếng Việt)



Source:

<https://www.geeksforgeeks.org/python-oops-concepts/>

Ví dụ:



Source: <https://hackmd.io/@yencheng/BksOTWNbd>

Object (đối tượng) :

- ❖ **Object** là một thực thể có trạng thái và hành vi (hành động, behavior):
 - State (trạng thái): các thuộc tính (attributes) của một đối tượng. Ví dụ:
 - Các trạng thái của Học Viên: tên, tuổi, điểm thi...
 - Các trạng thái của Xe Hơi: tên, năm sx, loại, kích thước, mã lực...
 - Behavior (hành vi, hành động): là các phương thức của một đối tượng
 - Identity (định danh):
 - Xác định tên duy nhất của một đối tượng
 - Giúp một đối tượng tương tác với các đối tượng khác

⇒ Object là khái niệm xuất phát từ thực tế, để xác định được class

Source: <https://hackmd.io/@yencheng/BksOTWNbd>

Class và các thuật ngữ đi kèm:

- ❖ **Class**: là prototype (của đối tượng) nhằm định nghĩa một tập các thuộc tính (attributes) giúp mô tả/phân định (characterize) các đối tượng của class đó.

Source: <https://hackmd.io/@yencheng/BksOTWNbd>

Instance (thể hiện):

- ❖ **Instance** là một đối tượng đơn lẻ của một class
 - Instantiation: tạo ra một instance

⇒ Instance là một object, nhưng là trong chương trình máy tính
("thể hiện" của object đó trong máy tính)

Source: <https://hackmd.io/@yencheng/BksOTWNbd>

Class bao gồm:

- ❖ Data member (thành phần dữ liệu): dữ liệu liên quan đến một class và các đt
 - Class variable: là biến được chia sẻ bởi mọi instance thuộc class đó
 - Instance variable: là biến của riêng instance, được định nghĩa trong các phương thức của instance đó
- ❖ Methods (phương thức): là một loại hàm đặc biệt được định nghĩa trong class

Source: <https://hackmd.io/@yencheng/BksOTWNbd>

Các thuật ngữ khác:

- ❖ **Inheritance** (kế thừa): là khả năng mà một class kế thừa các thuộc tính và phương thức từ class khác.

Gồm các loại:

- Lớp cơ sở
- Lớp dẫn xuất

NGƯỜI
Tên
Địa chỉ
Email
Số điện thoại
Hiển thị thông tin
Nhập thông tin

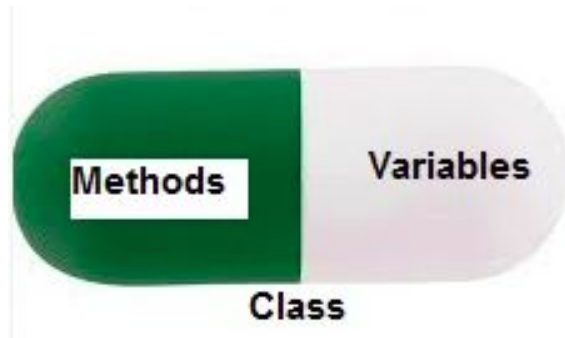
NHÂN VIÊN
Bằng cấp
Hiển thị thông tin
Nhập thông tin

KHÁCH HÀNG
Loại khách hàng
Hiển thị thông tin
Nhập thông tin

Source: <https://giasutinhoc.vn/lap-trinh/lap-trinh-c-sharp/tinh-ke-thua-trong-csharp-bai-6/>

Các thuật ngữ khác:

- ❖ **Encapsulation** (đóng gói):
gom các data và phương thức liên quan vào một class.



Các thuật ngữ khác:

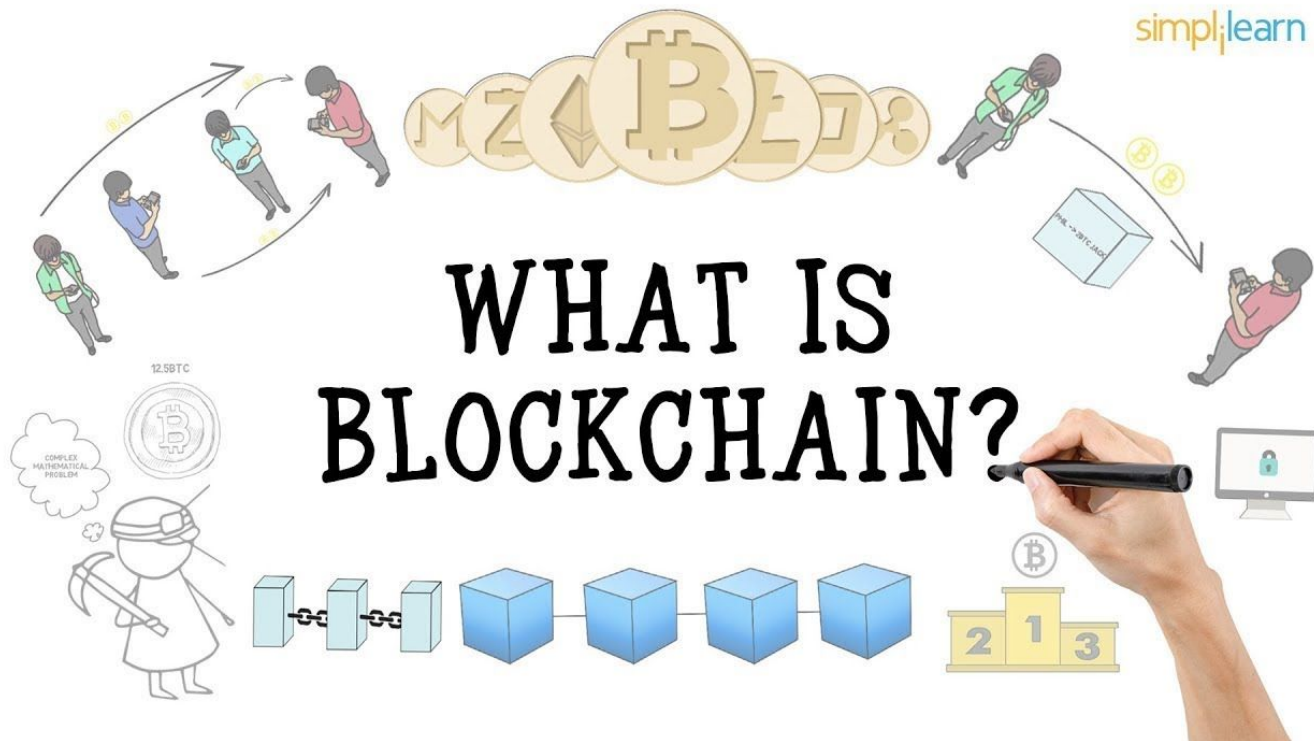
❖ **Polymorphism** (đa hình/đa xạ):

- Dạng compile-time (trong Python không thực hiện được): một lớp cho phép định nghĩa nhiều hàm cùng tên, nhưng khác tham số. Khi gọi hàm sẽ tùy theo các tham số mà chạy hàm nào.
- Dạng runtime: định nghĩa lại cùng một hàm thuộc lớp cơ sở ở các lớp dẫn xuất. Khi thực hiện gọi đối tượng (được định nghĩa từ trước theo tên của lớp cơ sở nhưng khi thực thi có thể là instance của các lớp dẫn xuất), tùy theo việc đối tượng thuộc lớp dẫn xuất nào mà chạy hàm tương ứng

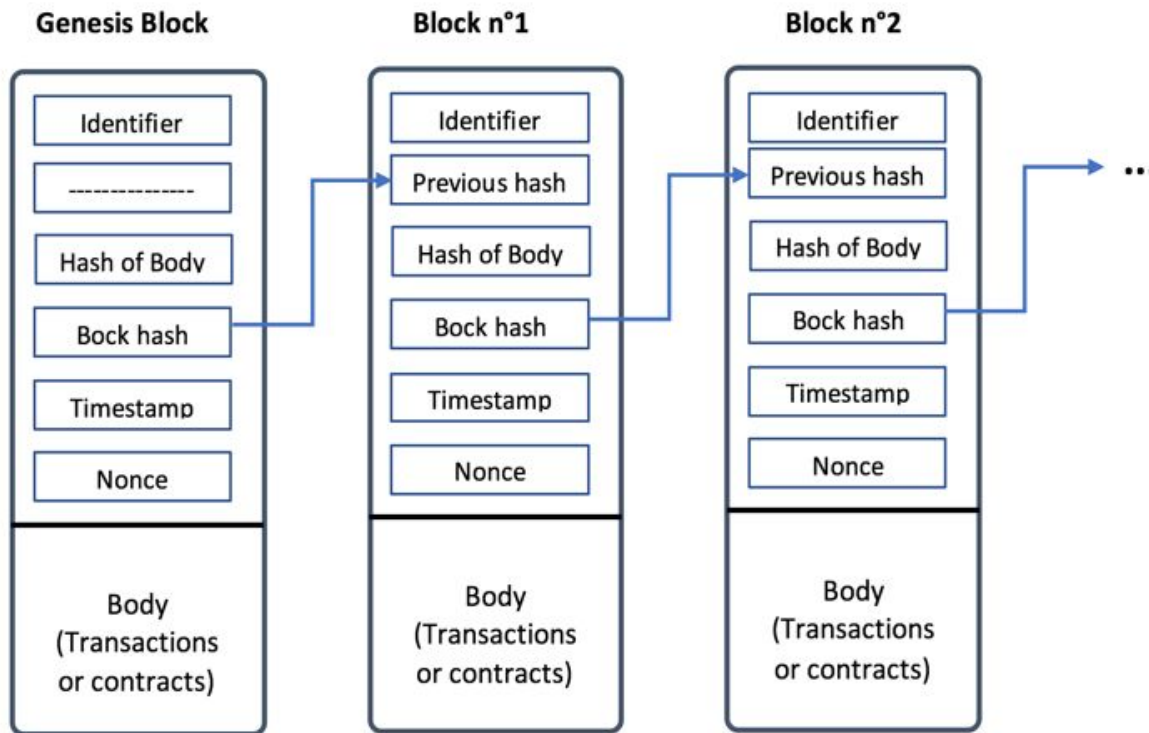
Tham khảo (tiếng Việt):

<https://openplanning.net/11417/python-inheritance-polymorphism>

Source: <https://hackmd.io/@yencheng/BksOTWNbd>



Source: <https://www.youtube.com/watch?v=RT7x0IQvSLk>



Source: <https://blog.eleven-labs.com/fr/bases-blockchain/>

Áp dụng Blockchain cho Truy xuất nguồn gốc Thực phẩm



Áp dụng Blockchain cho Truy xuất nguồn gốc Nông sản



Áp dụng Blockchain cho Truy xuất nguồn gốc Nông sản

Hai nguyên tắc TXNG:

Trả lời 5 từ hỏi:

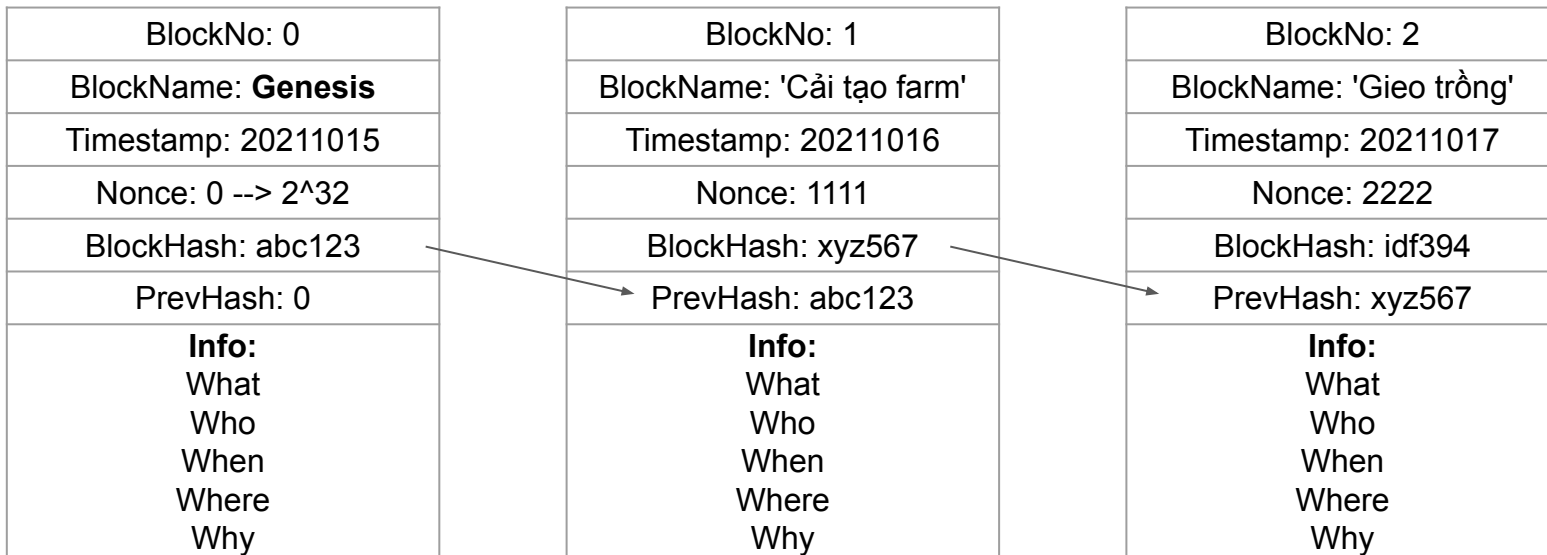
- ❖ What
- ❖ Who
- ❖ Where
- ❖ When
- ❖ Why

Nguyên tắc "một bước trước - một bước sau"



Áp dụng Blockchain cho Truy xuất nguồn gốc Nông sản

Target No: 2^{196}



THANK YOU!

